IN THE SPECIFICATION

Please replace the paragraph beginning at line 1, page 2 with the following rewritten paragraph:

A real-time operating system (RTOS), e.g., VxWorks™, OSE, Lynx, within a data processing system ~~is basic software for the tool management of the hardware and the software of the system and~~ provides hardware resource management functions, e.g., memory management, task management, and data input/output management.~~;~~ ~~and a~~ The operating system may further include a user interface for handling a screen display and the manipulation of a mouse. ~~An~~ The operating system may further include a module such as a timer management program, i.e., timer management system, for managing a plurality of timers that are started, stopped, idled, etc. by an application program in a data processing system. Application programs, e.g., word processing software, database software, software for calculation for tables, reside on top of the OS in the topmost layer of a hierarchial software arrangement.

Please replace the paragraph beginning at line 20, page 6 and ending at line 17, page 7 with the following rewritten paragraph:

Figure 1 illustrates a typical hardware configuration of data processing system 13 which is representative of a hardware environment for practicing the present invention. Data processing system 13 has a central processing unit (CPU) 10, such as a conventional microprocessor, coupled to various other components by system bus 12. A real-time operating system 40, e.g., VxWorks™, OSE, Lynx, runs on CPU 10 and provides control and coordinates the function of the various components of Figure 1. As stated in the Background Information section, a timer management program, i.e., timer management system, may reside in a module within the operating system 40. In another embodiment, an application 42, e.g., timer management system, may run in conjunction with operating system 40 and provide output calls to operating system 40 which implements the various functions to be performed by application 42. Read only memory (ROM) 16 is coupled to system bus 12 and

includes a basic input/output system ("BIOS") that controls certain basic functions of data processing system 13. Random access memory (RAM) 14, I/O adapter 18, and communications adapter 34 are also coupled to system bus 12. It is noted that software components including operating system 40 and application 42 are loaded into RAM 14 which is the computer system's main memory. I/O adapter 18 may be a small computer system interface ("SCSI") adapter that communicates with disk units 20 and ~~tape drives 40~~ tape drives 50. Communications adapter 34 interconnects bus 12 with an outside network enabling data processing system 13 to communication with other such systems. Input/Output devices are also connected to system bus 12 via a user interface adapter 22 and a display adapter 36. A display monitor 38 is connected to system bus 12 by display adapter 36. In this manner, a user is capable of inputting to system 13 through a keyboard 24 or a mouse 26 and receiving output from system 13 via display 38.

A2

Please replace the paragraph beginning at line 10, page 10 with the following rewritten paragraph:

Figure 3 illustrates an embodiment of the present invention of a state machine 300 which may be used in the operation of timer management system 200 (see Figure 2). State machine 300 depicts a timer management system 200 that manages timers for both a single task system, i.e., synchronous system, and a multi-task system, i.e., asynchronous system, concurrently. State machine 300 includes a single-task state machine 301 and a multi-task state machine 302. The states of the single-task state machine 301 comprise states 303, 304 and 305. The states of the multi-task state machine 302 comprise states 306, 307, 308 and 309.

A3

Please replace the paragraph beginning at line 17, page 10 and ending at line 2, page 11 with the following rewritten paragraph:

Referring to Figure 3, in conjunction with Figure 2, state machine 300 remains in state 303, non-existent ("NE"), until a timer is created. That is, state 303 represents a state in which a timer is non-existent. When a timer is created ("C"), state machine

A4

A4

300 transitions from state 303 to state 304. State 304 represents a state in which a timer is created and ready to be initialized. That is, the timer is idle ("I"). In one embodiment, timer services 240 creates a timer out of a block of memory. The block of memory allocated for creating the timer may then be stored in timer database 230. A description of the block of memory allocated for creating the timer is provided below in connection with the data structure of the time-out message.

Please replace the paragraph beginning at line 12, page 11 and ending at line 2, page 12 with the following rewritten paragraph:

A5

As stated above, a time-out message sent is the same block of memory as the block of memory allocated when the timer was created. The reason is that the procedure that creates the timer creates extra memory space for use by timer services 240 which is not seen by application 210. A drawing of an embodiment of a data structure 400 of the time-out message is illustrated in Figure 4. Referring to Figure 4, a ~~A~~ time-out message may comprise at least two distinct parts. An application part 410 may comprise information related to application 210. In a single-task system, the application part 410 may comprise the time-out function and a context field which are owned by application 210. In a multi-task system, the application part 410 may comprise the same context field as well as information necessary to identify application 210, e.g., queue to send the message, application task identification. A timer services part 420 may comprise some additional fields, e.g., pointers to memory addresses of timers stored in timer database 230. The timer services part 420 is not seen by application 210.

Please replace the paragraph beginning at line 3, page 12 with the following rewritten paragraph:

A6

Referring to Figure 3, in conjunction with Figure 2, application 210 may erroneously believe that the timer is operating at a state in which it is not. For example, application 210 may mistakenly believe that the timer is running when the timer is in actuality idle at state 304. Application 210 may then stop the timer at the

A4

idle state which is represented by the arched line at state 304. Furthermore, application 210 may mistakenly believe that the timer is idle when the timer is in actuality running at state 305. Application 210 may then activate the timer at the running state which is represented by the arched line at state 305.

Please replace the paragraph beginning at line 9, page 14 with the following rewritten paragraph:

A7

If while at state 307, asynchronous application 210 deletes the timer, then state machine 300 transitions from state 307 to state 308. When the time-out message is dequeued, the handle function transfers state machine 300 from state 308 in the multi-task state machine 302 to the non-existent state 303 in the single-task state machine 301. If while at state 307, asynchronous application 210 activates the timer, then state machine 300 transitions from state 307 to state 309. When the time-out message is dequeued, the handle function transparent to asynchronous application 210 transfers state machine 300 from state 309 in the multi-task state machine 302 to the running state 305 in the single-task state machine 301. It is noted that when the timer is reinitialized, i.e., restarted, in state 309, the timer uses the same allocated system memory as the system memory allocated when the timer was created in state 304.